Keymap.h - Test Output

Running `Source.h` will invoke the `KeymapTest` class across multiple distinct datatypes. Below is a copy of the console output after running this test process.

```
 MAP VALUE TYPE: char

        TESTING: Checking map is blank
       FUNCTION: isEmpty
EXPECTED RESULT: True
         RESULT: 1
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting KEY insert using nullptr keyword
       FUNCTION: insert(nullptr,v)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE and KEY insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using values
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Attempting to reinsert data
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Attempting to reinsert data using pointers
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Run the iterator forward
       FUNCTION: Iterator
EXPECTED RESULT: 1, 2, 3 (etc)
         RESULT: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Run the iterator backward
       FUNCTION: Iterator
```

```
EXPECTED RESULT: (etc), 3, 2, 1
         RESULT: [10:J], [9:I], [8:H], [7:G], [6:F], [5:E], [4:D], [3:C], [2:B]
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Printing each value
       FUNCTION: getValue(K)
EXPECTED RESULT: Each value is printed to the console.
         RESULT: [[A], [B], [C], [D], [E], [F], [G], [H], [I], [J]
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Updating a pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Value at position 0 will be updated
     MAP BEFORE: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],
      MAP AFTER: [1:J], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],

        TESTING: Removing a pair
       FUNCTION: removePair(K)
EXPECTED RESULT: Pair at position 0 will be removed
     MAP BEFORE: [1:J], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Updating a non-existing pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Before and after remain the same.
     MAP BEFORE: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Updating/Inserting a non-existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: A new pair will be inserted at the end
     MAP BEFORE: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Default value
         RESULT: J
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Updating/Inserting an existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: The new pair should be updated
     MAP BEFORE: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Actual value
         RESULT: A
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Get capacity
       FUNCTION: getCapacity()
EXPECTED RESULT: Capacity >= number of elements
         RESULT: 16
      MAP AFTER: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]

        TESTING: Reset the Map
       FUNCTION: reset()
```

```
EXPECTED RESULT: Map should be empty
     MAP BEFORE: [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I], [10:J]
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using pointers
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [1:A], [2:B], [3:C], [4:D], [5:E], [6:F], [7:G], [8:H], [9:I],


---------------------------------------------------

   MAP KEY TYPE: class std::basic_string<char,struct std::char_traits<char>,cla
 MAP VALUE TYPE: bool

        TESTING: Checking map is blank
       FUNCTION: isEmpty
EXPECTED RESULT: True
         RESULT: 1
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting KEY insert using nullptr keyword
       FUNCTION: insert(nullptr,v)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE and KEY insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using values
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Attempting to reinsert data
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Attempting to reinsert data using pointers
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Run the iterator forward
       FUNCTION: Iterator
```

```
EXPECTED RESULT: 1, 2, 3 (etc)
         RESULT: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Run the iterator backward
       FUNCTION: Iterator
EXPECTED RESULT: (etc), 3, 2, 1
         RESULT: [fozz:1], [fuzz:0], [bizz:1], [buzz:0], [fizz:0], [far:1], [bo
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Printing each value
       FUNCTION: getValue(K)
EXPECTED RESULT: Each value is printed to the console.
         RESULT: [[0], [1], [1], [0], [1], [0], [0], [1], [0], [1]
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Updating a pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Value at position 0 will be updated
     MAP BEFORE: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0
      MAP AFTER: [foo:1], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0

        TESTING: Removing a pair
       FUNCTION: removePair(K)
EXPECTED RESULT: Pair at position 0 will be removed
     MAP BEFORE: [foo:1], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Updating a non-existing pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Before and after remain the same.
     MAP BEFORE: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Updating/Inserting a non-existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: A new pair will be inserted at the end
     MAP BEFORE: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Default value
         RESULT: 1
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Updating/Inserting an existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: The new pair should be updated
     MAP BEFORE: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Actual value
         RESULT: 0
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Get capacity
       FUNCTION: getCapacity()
```

```
EXPECTED RESULT: Capacity >= number of elements
         RESULT: 16
      MAP AFTER: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:

        TESTING: Reset the Map
       FUNCTION: reset()
EXPECTED RESULT: Map should be empty
     MAP BEFORE: [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0], [bizz:
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using pointers
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [foo:0], [bar:1], [baz:1], [boo:0], [far:1], [fizz:0], [buzz:0


---------------------------------------------------

   MAP KEY TYPE: class std::basic_string<char,struct std::char_traits<char>,cla
 MAP VALUE TYPE: class std::basic_string<char,struct std::char_traits<char>,cla

        TESTING: Checking map is blank
       FUNCTION: isEmpty
EXPECTED RESULT: True
         RESULT: 1
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting KEY insert using nullptr keyword
       FUNCTION: insert(nullptr,v)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE and KEY insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using values
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f

        TESTING: Attempting to reinsert data
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f

        TESTING: Attempting to reinsert data using pointers
       FUNCTION: insert(K,V)
```

```
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f

        TESTING: Run the iterator forward
       FUNCTION: Iterator
EXPECTED RESULT: 1, 2, 3 (etc)
         RESULT: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f

        TESTING: Run the iterator backward
       FUNCTION: Iterator
EXPECTED RESULT: (etc), 3, 2, 1
         RESULT: [fozz:fozz], [fuzz:fuzz], [bizz:bizz], [buzz:buzz], [fizz:fizz
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f

        TESTING: Printing each value
       FUNCTION: getValue(K)
EXPECTED RESULT: Each value is printed to the console.
         RESULT: [[foo], [bar], [baz], [boo], [far], [fizz], [buzz], [bizz], [f
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f

        TESTING: Updating a pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Value at position 0 will be updated
     MAP BEFORE: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f
      MAP AFTER: [foo:fozz], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:

        TESTING: Removing a pair
       FUNCTION: removePair(K)
EXPECTED RESULT: Pair at position 0 will be removed
     MAP BEFORE: [foo:fozz], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Updating a non-existing pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Before and after remain the same.
     MAP BEFORE: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Updating/Inserting a non-existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: A new pair will be inserted at the end
     MAP BEFORE: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Default value
         RESULT: fozz
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Updating/Inserting an existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: The new pair should be updated
     MAP BEFORE: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
```

```
EXPECTED RESULT: Actual value
         RESULT: foo
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Get capacity
       FUNCTION: getCapacity()
EXPECTED RESULT: Capacity >= number of elements
         RESULT: 16
      MAP AFTER: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz

        TESTING: Reset the Map
       FUNCTION: reset()
EXPECTED RESULT: Map should be empty
     MAP BEFORE: [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:fizz], [buzz
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using pointers
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [foo:foo], [bar:bar], [baz:baz], [boo:boo], [far:far], [fizz:f


---------------------------------------------------

   MAP KEY TYPE: double
 MAP VALUE TYPE: float

        TESTING: Checking map is blank
       FUNCTION: isEmpty
EXPECTED RESULT: True
         RESULT: 1
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting KEY insert using nullptr keyword
       FUNCTION: insert(nullptr,v)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE and KEY insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using values
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Attempting to reinsert data
       FUNCTION: insert(K,V)
```

```
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Attempting to reinsert data using pointers
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Run the iterator forward
       FUNCTION: Iterator
EXPECTED RESULT: 1, 2, 3 (etc)
         RESULT: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Run the iterator backward
       FUNCTION: Iterator
EXPECTED RESULT: (etc), 3, 2, 1
         RESULT: [1.10101:0.104755], [1.999:0.947547], [1.888:0.847547], [1.777
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Printing each value
       FUNCTION: getValue(K)
EXPECTED RESULT: Each value is printed to the console.
         RESULT: [[0.147547], [0.247547], [0.347547], [0.447547], [0.547547], [
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Updating a pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Value at position 0 will be updated
     MAP BEFORE: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0
      MAP AFTER: [1.111:0.104755], [1.222:0.247547], [1.333:0.347547], [1.444:0

        TESTING: Removing a pair
       FUNCTION: removePair(K)
EXPECTED RESULT: Pair at position 0 will be removed
     MAP BEFORE: [1.111:0.104755], [1.222:0.247547], [1.333:0.347547], [1.444:0
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Updating a non-existing pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Before and after remain the same.
     MAP BEFORE: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Updating/Inserting a non-existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: A new pair will be inserted at the end
     MAP BEFORE: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Default value
         RESULT: 0.104755
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Updating/Inserting an existing pair
       FUNCTION: insertOrUpdate(K,V)
```

```
EXPECTED RESULT: The new pair should be updated
     MAP BEFORE: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Actual value
         RESULT: 0.147547
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Get capacity
       FUNCTION: getCapacity()
EXPECTED RESULT: Capacity >= number of elements
         RESULT: 16
      MAP AFTER: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0

        TESTING: Reset the Map
       FUNCTION: reset()
EXPECTED RESULT: Map should be empty
     MAP BEFORE: [1.222:0.247547], [1.333:0.347547], [1.444:0.447547], [1.555:0
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using pointers
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [1.111:0.147547], [1.222:0.247547], [1.333:0.347547], [1.444:0


---------------------------------------------------

   MAP KEY TYPE: float
 MAP VALUE TYPE: double

        TESTING: Checking map is blank
       FUNCTION: isEmpty
EXPECTED RESULT: True
         RESULT: 1
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting KEY insert using nullptr keyword
       FUNCTION: insert(nullptr,v)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting VALUE and KEY insert using NULL keyword
       FUNCTION: insert(k,nullptr)
EXPECTED RESULT: Empty map
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using values
       FUNCTION: insert(K,V), isEmpty()
```

```
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

        TESTING: Attempting to reinsert data
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

        TESTING: Attempting to reinsert data using pointers
       FUNCTION: insert(K,V)
EXPECTED RESULT: Set should return the same as above, with no additions
     MAP BEFORE: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

        TESTING: Run the iterator forward
       FUNCTION: Iterator
EXPECTED RESULT: 1, 2, 3 (etc)
         RESULT: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

        TESTING: Run the iterator backward
       FUNCTION: Iterator
EXPECTED RESULT: (etc), 3, 2, 1
         RESULT: [0.104755:1.10101], [0.947547:1.999], [0.847547:1.888], [0.747
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

        TESTING: Printing each value
       FUNCTION: getValue(K)
EXPECTED RESULT: Each value is printed to the console.
         RESULT: [[1.111], [1.222], [1.333], [1.444], [1.555], [1.666], [1.777]
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

        TESTING: Updating a pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Value at position 0 will be updated
     MAP BEFORE: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754
      MAP AFTER: [0.147547:1.10101], [0.247547:1.222], [0.347547:1.333], [0.447

        TESTING: Removing a pair
       FUNCTION: removePair(K)
EXPECTED RESULT: Pair at position 0 will be removed
     MAP BEFORE: [0.147547:1.10101], [0.247547:1.222], [0.347547:1.333], [0.447
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Updating a non-existing pair
       FUNCTION: updatePair(K,V)
EXPECTED RESULT: Before and after remain the same.
     MAP BEFORE: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Updating/Inserting a non-existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: A new pair will be inserted at the end
     MAP BEFORE: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
```

```
EXPECTED RESULT: Default value
         RESULT: 1.10101
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Updating/Inserting an existing pair
       FUNCTION: insertOrUpdate(K,V)
EXPECTED RESULT: The new pair should be updated
     MAP BEFORE: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Getting the value if exists, or a default value
       FUNCTION: getValueOrDefault(K,default)
EXPECTED RESULT: Actual value
         RESULT: 1.111
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Get capacity
       FUNCTION: getCapacity()
EXPECTED RESULT: Capacity >= number of elements
         RESULT: 16
      MAP AFTER: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754

        TESTING: Reset the Map
       FUNCTION: reset()
EXPECTED RESULT: Map should be empty
     MAP BEFORE: [0.247547:1.222], [0.347547:1.333], [0.447547:1.444], [0.54754
      MAP AFTER: The keymap is empty, and as such no printable values are store

        TESTING: Attempting to populate map using pointers
       FUNCTION: insert(K,V), isEmpty()
EXPECTED RESULT: Populated array
     MAP BEFORE: The keymap is empty, and as such no printable values are store
      MAP AFTER: [0.147547:1.111], [0.247547:1.222], [0.347547:1.333], [0.44754

---------------------------------------------------


D:\repositories\arraymap\arraymap-cpp\x64\Debug\arraymap-cpp.exe (process 13124
To automatically close the console when debugging stops, enable Tools->Options-
Press any key to close this window . . .
```